

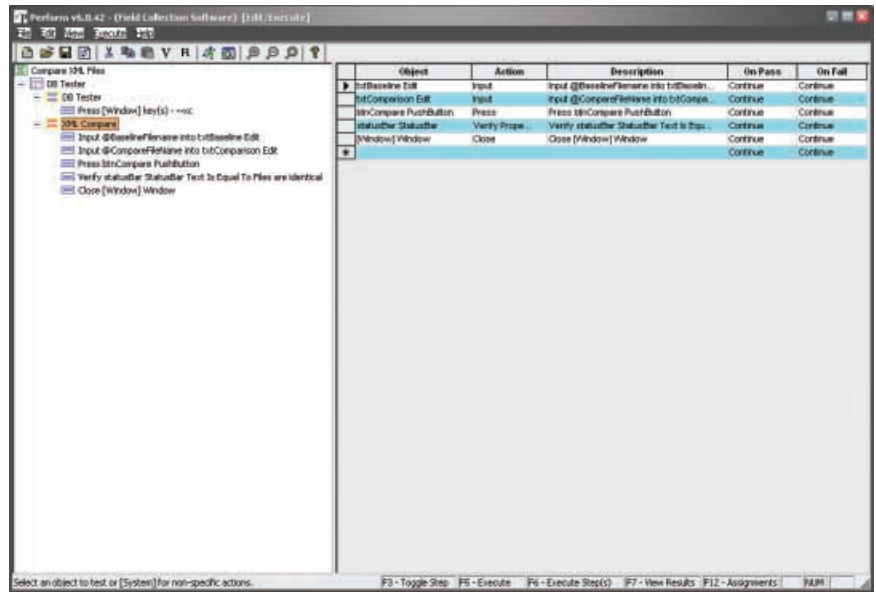
A Look at Worksoft Certify

by Geoff Stewart

EARLY IN 2003, OUR AUTOMATION TEAM was faced with an interesting challenge. Our current tool had been used successfully on multiple projects. Unfortunately, each project team had used the tool in different ways so it was difficult to share resources across projects. The tool, like most others, was dependent on knowing how to write scripts using a proprietary language. Based on these difficulties, our team was given a goal: to implement a solution that would be used consistently across projects and also enable employees with no scripting experience to automate tests. This would free up our automation team members to focus on writing test harnesses and simulators.

We did a build vs. buy evaluation and, given our small automation staff, decided to try to find a vendor that met our requirements. Buying would also prevent us from being stuck with maintaining and enhancing whatever we developed on our own. For the vendor evaluation, we looked at Worksoft Certify, a competing vendor, and an open source solution. Worksoft Certify's major selling point was the relational repository it uses for all test assets. This central database contains all test requirements, test cases, test data, and application objects, facilitating reuse of existing tests and simplifying maintenance. At the time, the competing vendor was tightly coupled to a playback tool for which we did not own licenses, and we didn't want to add another major vendor to the picture. The open source solution involved creating scripts using spreadsheets, which are simple to create, but we felt resulted in long-term maintenance issues. Overall, it appeared Worksoft Certify would be the best fit for our organization, so we decided to proceed with a pilot project.

The goal of our pilot project was to enable more people to easily automate tests. For the pilot, we selected an appli-



Worksoft Certify's major selling point for Stewart was the relational repository it uses for all test assets.

cation that previously had been automated with another tool. The chosen application required such detailed scripting knowledge that, before the pilot began, only one person could support it. After some initial setup described later in this article, all three members of the pilot team were able to use Certify to automate tests using only the existing test specification. Based on this experience, Certify has proved to me to be a viable solution for our automation needs.

Certify Architecture

Certify is made up of three components: the GUI client, the database server, and a function library. The GUI enables users to plan, design, execute, and maintain tests. The central database server can be Access, SQL Server, or Oracle. The function library is custom code that implements the necessary actions for different classes (Pushbutton, TextBox). The library is specific to the development platform of the application under test (.NET,

VB, Java, PowerBuilder). These libraries can be purchased from Worksoft or developed on your own.

Depending on your development platform, you may still need an application mapping/playback tool in addition to Certify. The mapping tool needs to generate an application map in a predefined format that Certify can import to the database. The playback tool is called by the function library and handles execution of test steps. Worksoft offers native solutions for some platforms (Mainframe, HTML, Java) and has an open interface that will interact with most of the other major tool vendors. For our .NET applications we chose to use TestComplete for mapping and TestExecute for playback. Both of these solutions are developed by AutomatedQA and are relatively inexpensive (\$350 and \$50 respectively).

What Does Certify Do?

Worksoft Certify enables nonprogrammers to automate tests. Certify differs

from traditional automation tools because it allows users to design and automate tests at the same time. Certify's major advantage over scripting is that it allows subject matter experts who are more familiar with the business domain to automate tests. On the downside, the lack of scripting limits your ability to do things such as file parsing. Certify is a huge improvement over capture/playback because tests are easy to maintain and drive with different sets of data. The only advantage capture/playback has over Certify is quicker generation of tests, as there is no initial infrastructure setup required.

Users design tests by specifying the application under test, the application window, and the steps to perform against that window. There are three main user entries when adding a design step to your test. First, you select a valid object from a drop-down list for the current application window. After this selection is made, the second column contains a drop-down list displaying valid actions for the selected object. The third column defines any parameters that need to be used as part of the action. These parameters can be expressed as variables, which allows you to data drive your test for different conditions. In addition to these three values, Certify also provides error handling based on the results of the step. Once the application is mapped and imported into the database, anyone, regardless of programming experience, can design and automate tests.

One of Certify's most valuable functions is the application map import process. This process determines whether a map contains new or modified objects. The import process looks for the application maps at a user-specified location. During import, all maps are compared

One of Certify's most valuable functions is the application map import process. This process determines whether a map contains new or modified objects.

against what is currently in the database. After import, objects are marked as New, Deleted, Modified, or Unchanged. If an object has been modified or deleted and is being used as part of existing tests, the user is forced to resolve the change. The UI allows you to find/replace the old object with the new object. This is a critical step that makes it much easier to maintain tests between two versions of your application.

In addition to automated testing, Certify includes several other useful features. A base set of Crystal Reports for running management reports and analyzing results is included with the client. You can also easily create or import requirements and link them to your tests. Users can use Certify for designing manual tests. Manual tests use the same designer as automated tests but during execution are displayed as step-by-step instructions for running the test.

What Can Go Wrong?

Certify, for all its potential, is still not the silver bullet tool that satisfies every test scenario. The product is relatively new, so the user interface can seem a bit buggy at times and isn't as feature rich as other vendors' products. However, from our perspective this was a benefit because our previous tool had a ton of functionality that was never used. To quote my boss, "When evaluating available functionality in a product, it is simply a cost-to-value relationship. When calculating cost, it is important to understand how unused functionality in a product can substantially increase the cost of product implementation. In the case of Certify, we are using approximately 90 percent of the functionality.

This allows our team to target high-value tool-usage areas and minimize the cost of more complex solutions."

If you implement Certify to sit on top of a playback tool, it inherits any existing limitations of that tool. The biggest issue that all test tools suffer from is custom controls. In our case, our developers use a third-party tool, which has created issues with automating tab pages and datagrids. Certify provides a hook for writing custom code to handle these situations, but it is not something that works straight out of the box. Although Certify does require this extra effort, I don't believe any tools handle custom controls without some additional work.

A major selling point of Certify is the fact that there is no scripting involved in designing and automating tests. The downside of this for us has been that it's harder to script your way around more complex testing scenarios, such as testing an application with no UI. Additionally, because this is primarily an automated test tool, the current implementation for manual testing is a bit weak. We have a couple of groups that do mostly manual testing, but they are hesitant to switch to Certify until some enhancements are made.

Closing

Overall, I am a huge advocate of this product. Certify has demonstrated the ability to enable more team members to automate tests. On my current project, this has freed me up to design and implement several tools that otherwise would have been put off until a later date. If your group is debating moving toward a next generation test solution, Worksoft Certify is definitely worth a look. **{end}**

Geoff Stewart helps test and development teams improve their test driven development activities by creating automated tools and simulators for Itron, Inc. His innovation with automated tooling ensures the successful test of projects.

Sticky Notes

For more on the following topics go to www.stickyminds.com/bettersoftware

■ More on Worksoft and AutomatedQA

The opinions stated in "Tool Look" are solely those of the author. We accept unsolicited product reviews written by actual users of the tools only. We do not accept reviews written by the vendor, vendor affiliates, or vendor subsidiaries. To be considered for publication, send your "Tool Look" to editors@bettersoftware.com.